Article

A Device Anonymization Protection Method Based on Address Hopping

Bo Zhang , Zesheng Xi , Chuan He , Yunfan Wang * , Tao Zhang

Posted Date: 7 March 2025

doi: 10.20944/preprints202503.0526.v1

Keywords: device anonymization protection; address hopping; mobile target defense (MTD); software defined networking (SDN); backtracking method

Article

A Device Anonymization Protection Method Based on Address Hopping

Bo Zhang [1,†], Zesheng Xi [2,†], Chuan He [3,†], Yunfan Wang [4,†] and Tao Zhang [4,*]

[1] State Grid Laboratory of Power Cyber-Security Protection and Monitoring Technology, China Electric Power Research Institute Co., Ltd., Nanjing, China; School of Cyber Science and Engineering, Southeast University, Nanjing, China
[2] Affiliation 2; zhangbo6@epri.sgcc.com.cn (B.Z.); 4317045xi@163.com (Z.X.); 230240027@seu.edu.cn (C.H.); 230240007@seu.edu.cn (Y.W.)
[*] Correspondence: zhangtao3@epri.sgcc.com.cn
[†] These authors contributed equally to this work.

Abstract: With the growth of IoT technology, connected devices have surged, increasing security risks, especially for devices lacking authentication. Anonymization protection prevents data leaks and control theft but traditional methods lack dynamism, struggle to balance privacy and availability, and remain vulnerable to targeted attacks. Anonymization protection techniques can prevent the leakage of sensitive information and the theft of control privileges, significantly improving the security of devices. However, traditional anonymity protection methods lack dynamism, making it difficult to trade-off between data availability and privacy protection, and attackers can discover system vulnerabilities through reconnaissance and analysis, leaving devices still vulnerable to targeted attacks. In this paper, we propose a device identity anonymization protection method based on address hopping, using the address hopping policy in the Mobile Target Defense (MTD) technique. It collects network topology and node state information, constructs a virtual network topology by backtracking method, and periodically replaces the paths and addresses under the satisfaction of specific constraints, so as to realize the anonymity of network devices. It effectively reduces the risk of device attacks, optimizes network performance, and maintains data availability by dynamically adjusting device addresses in the network. Experiments using Mininet and Ryu controllers show the approach significantly reduces host scans and data exposure compared to unprotected policies.

Keywords: device anonymization protection; address hopping; mobile target defense (MTD); software defined networking (SDN); backtracking method

1. Introduction

As the Internet and Internet of Things technology (IoT) advance rapidly, the number and variety of connected devices have experienced explosive growth, which has greatly enriched people's daily life and work style. However, while IoT introduces intelligence, it also poses numerous security and privacy challenges. The current power equipment market lacks stringent access control mechanisms and unified technical management standards, some equipment lacks an identity authentication mechanism, which results in the exchange of sensitive information such as control commands that are easily attacked by the adversary, and then the attacker can issue malicious control commands to steal the control of the distribution network equipment terminals. The attacker can then issue malicious control commands to steal the control rights of the distribution network equipment terminals. Therefore, anonymous protection of equipment identity has become an increasingly important research area.

In the field of device anonymity protection technology, there are two main categories of methods. The first class of methods is to use the physical differences of the device itself to construct anonymity protection mechanisms, and this type of technology focuses on protecting the hardware characteristics of the device, such as the MAC address, the device serial number, and so on, and prevents the device from being recognized by its physical characteristics by hiding or modifying this information through technical means. The second type of method is to use the differences in network traffic generated by the device to construct anonymity protection measures. This type of technology focuses on protecting the network behavioral patterns of the device, and changes the network traffic characteristics of the device by means of encryption, proxies, obfuscation, etc., so that even if the device is active in the network, its identity information is not easy to be traced and identified.

However, these methods have the following two limitations: (1)**Staticity and predictability** :

Traditional methods are often based on static data models, which generalize or suppress individual attributes in a dataset to reduce the risk of individuals being identified. However, these methods do not take into account the changes or dynamics of the data over time and lack the ability to adjust dynamically, making it possible for an attacker to predict the data patterns after anonymization through long-term observation and analysis.

(2) **Trade-offs between data availability and privacy protection**: In order to protect privacy, traditional anonymization techniques may sacrifice the detail and usefulness of the data, making it difficult to use the data for further analysis and research after desensitization.

To overcome these constraints, this thesis introduces a unique identity anonymity protection method that relies solely on the Moving Target Defense (MTD) technique, with a special focus on the address hopping strategy. The MTD technique enhances system security by dynamically altering its configuration, thereby making it more challenging for attackers to comprehend and exploit.

The main contributions of this paper are:

(1) In order to improve the dynamics and anti-prediction of the system, this paper introduces the MTD technique, which realizes dynamic changes at the network layer through the end information dynamic hopping technique, the routing dynamic hopping technique and the topology dynamic defense technique. By dynamically changing the device addresses in the network, the risk of being attacked is effectively reduced and the network performance is optimized. This method effectively overcomes the problem of static nature of traditional methods, which makes it difficult for attackers to predict device identities through fixed patterns.

(2) In order to optimize the balance between data availability and privacy protection, the MTD technique protects data privacy by dynamically changing the network configuration, making it difficult for attackers to continuously track and analyze the data. This approach maintains data availability in a changing environment because it allows data to be accessed and used efficiently while maintaining privacy.

(3) In this method, effectiveness analysis, data forwarding volume analysis, and availability analysis are performed by conducting experiments in a simulated network environment constructed by Mininet network simulation platform and Ryu controller. The experimental outcomes demonstrate that the proposed method can effectively decrease the likelihood of the host being targeted by scanning attempts, reduce the amount of data acquired by the attacker, and reduce the communication delay by about 60% compared to the unprotected strategy under different attack intensities, as well as reduce the CPU load accordingly when the hopping frequency is reduced.

2. Related Works

Deep learning-based device identity anonymization protection techniques focus on the problem of how to protect the identity information of a device without compromising the user's privacy. This problem is particularly important in areas such as the Internet of Things (IoT) and smart devices, which widely collect and transmit personal data that can be misused or leaked if not protected.

Deep learning techniques, particularly Convolutional Neural Networks (CNN)and Residual Networks (RNN), have attained significant accomplishments in the fields of image recognition and speech recognition [1,2]. For example, better identity anonymization is achieved by constructing an identity decoupling network that decouples identity from other attributes using conditional multiscale reconstruction loss (CMR) and identity loss. In addition, there have been studies on generating anonymized identity vectors by changing the distortion angle in the angle space to directly control the identity of anonymized faces. In addition to this, feature extraction and classification of signals emitted from devices by deep neural networks can be used to identify and protect the identity of devices [3]. Deep learning can also be used to generate adversarial samples to improve the anonymization of sensitive data such as face images to achieve better privacy protection under deep neural networks [4]. These methods not only solve the privacy leakage problem, but also promote the overall development of AI technology and play an important role in national political security, economic security, social security and cyber security.

In terms of specific implementations of device identity anonymity protection, some studies have proposed methods such as the Wearable Device Data Privacy Preserving Transform based on Distinguishing Self-Encoder(Dis AE)and the Wearable Device Data Privacy Preserving Transform based on Chunked Discrete Cosine Transform and Self-Encoder(BDCT-AE) [5]. These methods reduce the identification risk of wearable device data by adding noise to the data or changing the representation of the data to make identification more difficult.

Wang Z et al. [6] proposed a scheme to effectively protect users' location privacy by penalizing location leakage and spoofing. The basic idea is that when sending a location-based service (LBS) query request, the user first constructs an anonymized zone by obtaining at least the real locations of other collaborating users, and then submits the anonymized zone to the LBS provider instead of his/her own real location, thus effectively protecting personal location privacy. In addition, compared with traditional schemes, this scheme reduces the computation latency and communication overhead of the requesting and collaborating users, while safeguarding against location leakage and location spoofing during the construction of anonymous zones, without the need for a third-party entity.

Wang W [7] proposed a smartphone authentication method based on optimized convolutional deep belief network (CRBM). The system uses two CRBM pooling layers and integrates features through GAP processing and RMS connectivity layers to reduce network parameters and shorten training time. Subsequently, a backpropagation algorithm is introduced to perform supervised learning of the model parameters and regulate the weights between the RMS connectivity layer and the output layer to determine the model's user class. Finally, Softmax classifier is added and ACC, FRR and FAR is selected as a metric to assess the precision of the algorithm and establish the accuracy of user authentication.

The study by Zou Z [8] and others reviews recent advances in the field of face anonymization, focusing on privacy protection during deep learning. Although Generative Adversarial Networks (GAN) perform well in face anonymization, there is still room for improvement, such as how to avoid information leakage and absolute privacy protection. In addition, the paper explores the question of whether wearing masks can effectively protect personal privacy and proposes a benchmarking tool to assess privacy invasion. In terms of face video anonymization, the article introduces two approaches, focusing on facial region privacy and biometric-oriented privacy.

Yao H et al. [9] proposed a differential privacy-preserving algorithm for deep neural networks that utilizes Funk-SVD matrix decomposition and gradient reconstruction to eliminate redundant noise. This method can effectively prevent background attacks, such as the Netflix privacy leakage incident, and solves the difficulties existing in the fusion of traditional anonymization methods with deep learning.

Tong X et al. [10] have proposed various anonymity detection and privacy preservation techniques in federated learning (FL). These techniques include combining homomorphic encryption and differential privacy to protect client data, proposing efficient and secure aggregation methods, and a learning framework based on multi-party computation. In addition, the paper discusses the defense measures in the case of non-independent homomorphic distribution in FL to cope with the problem of uneven distribution of client data in real environments.

Lu B et al. [11] proposed a direct anonymous authentication scheme that can be used to anonymize the identity of the end device through a trusted chip. This scheme is used for any service provider without the need of coercive force to guarantee the implementation and protects the user privacy before it is violated. Also the authors give a security proof of the correctness, anonymity and unforgeability of the scheme under the stochastic predicator model.

In the authentication of IoT devices, based on elliptic curve cryptography, a lightweight method for anonymous authentication of devices can be proposed [12]. This method can reduce the computation and storage overhead while ensuring security, thus adapting to resource-constrained IoT environments. Cheng J et al. [13] studied the privacy protection and application of trajectory data in the context of mobile devices and location-based technologies. They proposed various techniques such as differential privacy, location coordinate transformation and cryptography based methods for protecting user's trajectory privacy.

These literatures demonstrate a variety of different techniques and approaches to protect the anonymity of device addresses, covering a wide range of aspects from blockchain technology to trusted chips, and from Hidden Markov Models to spatial and temporal masking. Each method has its unique advantages and application scenarios, providing a rich theoretical foundation and technical support for variable anonymity protection of device addresses.

However, although several techniques and methods have been proposed for protecting device identities, there are still some challenges and limitations. For example, how to improve the anonymization effect while minimizing the impact on device performance; how to design anonymization strategies that can resist Advanced Persistent Threats (APT) and other sophisticated attacks. Therefore, based on existing research, a deep learning-based device identity anonymization protection method is proposed. The method balances the anonymization effect and efficiency, and obtains the optimal hopping strategy based on the evolutionary game strategy.

3. Methodology

In modern network security, protecting the anonymity of device addresses is crucial. The address-hopping based device address variable anonymity protection method is effectively implemented in Software Defined Networking (SDN). This approach utilizes the Moving Target Defense (MTD) technique to dynamically adjust the device addresses in the network to protect the anonymity of network devices and enhance network security. First, the source host sends a packet to the head switch, which decides whether to forward the packet or send the information to the controller through flow table matching. After receiving the information, the controller selects the appropriate transmission path and virtual IP to hide the real device information, and generates the corresponding flow table rules to be sent to the switches in the path. The packets are forwarded among the switches according to the flow table rules until they reach the destination host. In order to maintain anonymity, the controller will replace the virtual IP and transmission path and reissue the flow table rules after a set time interval, so that the addresses and paths in the data flow change periodically, thus realizing the anonymization of the device identity, which not only improves the anonymity and security of the network, but also strengthens the network's dynamic adaptability and anti-attack capability. The framework diagram of the methodology is shown in Figure 1.

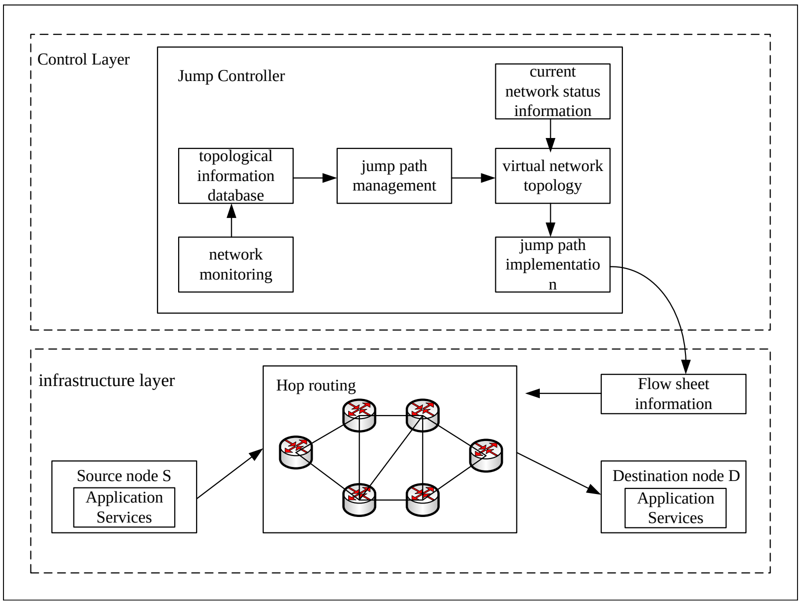

Figure 1. Address hopping based anonymity protection framework for SDN networks

3.1. Moving Target Defense Approach for Address Hopping

Moving target defense is an advanced network security strategy whose core idea is to confuse attackers by implementing continuous and dynamic changes to increase their attack cost and complexity and reduce their attack success rate. Moving target defense is not a specific defense method, but a design guideline that can be applied to a property of the protected system to derive a variety of specific defense mechanisms.

In traditional network communication, the IP addresses and ports of nodes are fixed, which enables attackers to determine the network location of a target and then launch an attack by obtaining it in advance or analyzing it through intercepted packets. To cope with this situation, the address hopping technique has emerged as an effective proactive defense, which is particularly suitable for countering traditional man-in-the-middle attacks, and it operates independently of the attacker's specific attributes.

Based on the concept of moving target defense, address hopping techniques dynamically alter the network locations of the two communicating parties, thus constantly invalidating the a priori knowledge of the attacker. Currently, the address hopping technique encompasses three primary methods of implementation: real IP address hopping, virtual IP address hopping, and port address hopping [14]. All of these methods significantly enhance the challenge for attackers in locating the target network, thereby bolstering the overall security of the system.

3.1.1. Real IP Address Hopping

Real IP address hopping entails dynamically modifying the IP address of either the client or server endpoint during packet transmission, based on system settings, as a means of mitigating man-in-the-middle attacks. There are mainly the following methods to realize real IP address hopping:

Dynamic Host Configuration Protocol (DHCP): The use of DHCP servers to periodically change the IP address of the terminal to achieve IP address dynamics, can effectively defend against the spread of worms. This method continuously requests new IP addresses through the DHCP server to realize IP address hopping.

Address Randomization Algorithm: The IP address is dynamically changed during the communication process through the address randomization algorithm, which uses an improved DHCP server and applies a hash value and a random number generator to produce random IP addresses for hopping.

3.1.2. Virtual IP Address Hopping

Establish a mapping between virtual and real IP addresses for the communication endpoints, and change the virtual IP address while the actual IP address remains unchanged when the address is hopped. There are several ways to realize virtual IP address hopping:

Mapping of virtual IP and real IP: By setting up a mapping of virtual IP and real IP addresses for communication endpoints, only the virtual IP address can be changed when the address is hopped, with the real IP address remaining static. This method can dynamically change a device's network location without interrupting existing network connections, making it harder for attackers to identify and launch attacks on specific devices.

Using virtual IPv6 addresses for packet transmission allows two hosts to communicate while minimizing packet loss caused by address hopping, thanks to the utilization of a sliding window mechanism. This method enables dynamic address changes while maintaining high speed data transmission.

3.1.3. Port Address Hopping

By dynamically changing the address and port information of both parties during communication, an attacker is prevented from intercepting the communication data exchanged between the client and the server. There are mainly the following methods to realize port address hopping:

Dynamic Address Translation (DyNAT) technology: By deploying the DyNAT client-side extension and configuring the DyNAT gateway on the server end, the DyNAT plug-in is responsible for processing request messages from the client and response messages into the client, and the DyNAT gateway is responsible for processing request messages arriving at the server side and response messages departing the server side [15].

A self-synchronization mechanism based on cryptographic hashing for port address hopping communication: this approach utilizes the HMAC mechanism to generate a message authentication code (MAC), which then serves as the synchronization data for encoding and decoding port addresses. This method offers a capability for port address hopping on a per-packet basis and ensures covert message authentication within the port address hopping system [16].

3.2. SDN-Oriented Mobile Target Defense Technology

3.2.1. Architecture of SDN

SDN, being a novel network architecture, has a technical architecture that is very different from the traditional network architecture [17]. Its core idea is to separate the control logic and forwarding logic of traditional network equipment, using centralized control to control network equipment, transferring the control logic to an ordinary computer independently into a controller. The control layer controls the data forwarding layer through standard interface protocols, which improves the controllability and programmability of network equipment. The openness to programming interfaces makes control of the network more agile and reduces the overhead cost of network control.

As shown in Figure 2, the SDN architecture is typically structured into three layers: the bottom layer is the infrastructure layer, which mainly consists of network components that support SDN functionality, and functions as the data plane within the SDN architecture; the middle layer is the control layer, where the main components are controllers controlling the SDN network, and is the control plane in the SDN architecture; and the top layer is the application layer, which consists of the SDN applications and services, and is the Application plane [18].

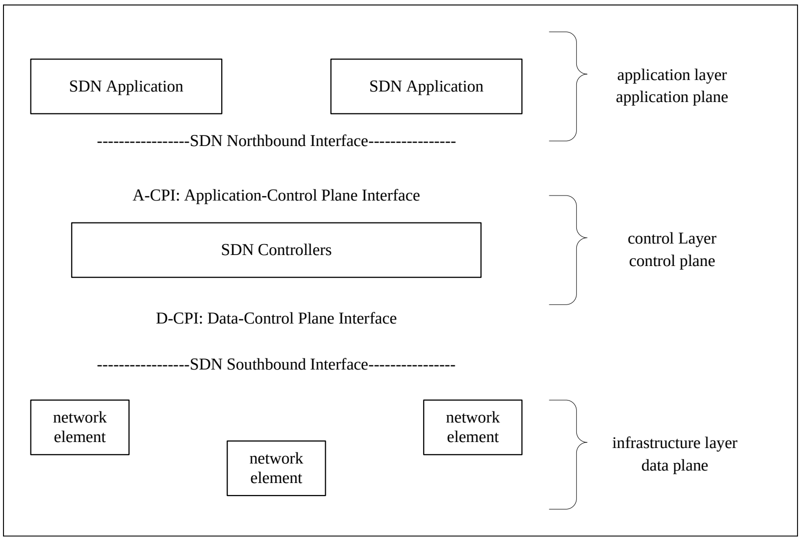
SDN Application
SDN Application
application layer
application plane
-----------------SDN Northbound Interface----------------
A-CPI: Application-Control Plane Interface
SDN Controllers
control Layer
control plane
D-CPI: Data-Control Plane Interface
-----------------SDN Southbound Interface----------------
network element
network element
network element
infrastructure layer
data plane

Figure 2. Schematic diagram of network architecture for SDN

3.2.2. Combination of Address Hopping Technology and SDN Technology

SDN technology provides strong support for address hopping technology, and the combination of the two not only enhances the dynamics and flexibility of the network, but also significantly improves the security of the network [19].

Programmability of SDN and Address Hopping Technology: The core advantage of SDN architecture is its centralized management of control, which allows the network to be manipulated and processed through the programming of SDN controllers, showing a high degree of programmability. In address-hopping technology, many critical operations need to be handled at the network level, and this feature of SDN facilitates the realization of these operations. By programming SDN controllers, we can flexibly define and implement address-hopping rules and policies, thus enhancing the dynamics and security of the network.

Centralized Management and Hopping Synchronization Policy: The centralized management capability of the SDN architecture not only facilitates network manipulation, but also facilitates the design and implementation of hopping synchronization policy in address hopping technology. the SDN controller can globally monitor and control all the data streams in the network, so that when address hopping is carried out, it can synchronize and update the address information of the matching streams in the transmission path. This centralized control mechanism ensures the continuity and consistency of the data streams during address hopping and reduces network disruption and confusion caused by address changes.

Address hopping in the form of data streams: In SDN, data packets are accepted and forwarded in the form of generalized streams, which provides new possibilities for address hopping techniques. By performing address hopping operations in the form of data streams, we can dynamically change the IP addresses of devices in the network without interrupting existing sessions. This approach not only improves the flexibility of the network, but also enhances the defense against attackers, who find it difficult to predict and track the real-time addresses of target devices.

Separation of the control plane from the data forwarding plane: The separation of the control plane from the data forwarding plane in the SDN architecture provides an additional security advantage to address-hopping techniques. Data streams flowing through the switch must go through the SDN controller’s decision making before being forwarded, which facilitates defense against SDN insider threats. During address hopping, the controller can monitor and adjust the data flow in real time to ensure that network traffic is forwarded securely and accurately, even when the address changes.

3.3. SDN-Based Address Hopping Model Design

First, the address hopping model collects key information about the underlying network, including network topology and node state information. This information is critical for understanding the current state of the network and potential path choices. The collected information is used to construct a global view of the network to provide data support for subsequent construction of paths. Second, the backtracking method is used to construct its corresponding virtual network topology for each possible data flow. In the process of constructing the virtual network topology, several constraints need to be considered to ensure the feasibility and efficiency of the selected paths. These constraints include reachability, transmission delay, link capacity, and non-overlap constraints [20].

3.3.1. Address Hopping Path Management

The module uses an algorithm based on the backtracking method to construct the virtual network topology. In constructing these paths, the module formalizes the constraints that the hopping paths should satisfy in terms of three key dimensions forwarding path capacity, transmission delay, and reachability. This ensures that the selected paths are not only able to carry the necessary data traffic, but also reach their destinations in a reasonable amount of time.

The core purpose of random routing is to increase the unpredictability of network communication by actively transforming the transmission paths between source and destination nodes, thus making it difficult for an attacker to intercept the complete communication content or block the communication between source and destination nodes. The effectiveness of this strategy relies on the path inconsistency, i.e., the forwarding paths between the source and destination nodes are different for each communication and there are no identical nodes other than the source and destination nodes. This inconsistency destroys any pattern recognition that an attacker may rely on, increasing the difficulty of the attack.

To further ensure the security of random routing transformations, a non-overlap constraint is used. This means that at each path hop, the new path does not share any intermediate nodes with the previous path. This constraint not only enhances the unpredictability of the paths, but also reduces the possibility of an attacker predicting future paths by recognizing path patterns, thereby enhancing the overall network security.

First, the network system is represented as an undirected graph $G = (V, E)$, where V is the set of forwarding nodes and E is the set of links containing a node $(v_1, v_2, \ldots, v_a)$ and containing b links $(e_1, e_2, \ldots, e_b)$, respectively. The set of nodes adjacent to node v_i is $\psi(v_i)$, the connection between node v_i and node v_j is represented by the link $e_{i,j}$, and the hop distance between node v_i and node v_j is $Link^{i \leftrightarrow j}$. Where $Link^{i \leftrightarrow j}$ contains n paths $(Link_1^{i \leftrightarrow j}, Link_2^{i \leftrightarrow j}, ..., Link_n^{i \leftrightarrow j})$, any path consists of a number of forwarding nodes v [21].

(1) Bandwidth limitation constraints

In order to avoid network link overload and packet loss, when constructing communication paths, it is necessary to impose a bandwidth constraint on the generation process of the routing hop space to ensure the reliability of data transmission of the generated paths. This constraint aims to guarantee the stability and efficiency of network transmission.

$$C(e_{i,j}) \geq C_t \tag{1}$$

where $C(e_{i,j})$ represents the transmission capacity of link $e_{i,j}$, C_t denotes the bandwidth threshold required for communication.

(2) Communication delay control

In order to satisfy the user's requirements for communication quality and ensure that communication delays remain within tolerable bounds, the generation of the routing hop space must take into account the constraints of the transmission delay. The primary factor influencing the new communication delay is the hop count along the routing path; thus, the constraint on transmission delay can be described in terms of the path's hop count.

$$D(v_j) \leq L_m ax \tag{2}$$

Where $D(v_j)$ denotes the depth of exploration when reaching a new node v_j, $L_m ax$ denotes the permissible maximum number of routing hops.

(3) Non-overlap constraint

In order to prevent the routing hopping mechanism from becoming predictable, strategies need to be adopted during the generation of the routing hopping space to avoid multiple occurrences of the same node (excluding source and destination nodes) in the path. For this purpose, the overlap between different paths should be fully considered and measures should be taken to minimize this overlap in order to enhance the unpredictability of routing hops.

$$v_j \neq L \tag{3}$$

Where L stands for the routing hop distance that has been determined. Regarding the non-overlapping constraint, it means that no nodes, except for the source and destination, repeat within the hop space.

3.3.2. Constructing Virtual Network Topology Based on Backtracking Method

Backtracking method is a kind of optimization search method. Its core idea is that in the exploration process, when exploring to a certain step, if the searched target is to satisfy the conditions of the selection of the best, then continue to move forward to search, or else return to the previous step, and re-select. The exploration behavior of the backtracking method follows the depth-first strategy, the root node serves as the initial expansion node, and a child node that has been explored is considered a potential new node. If this new node satisfies the selection criteria, it will be chosen as the next expansion node, and the exploration process will proceed. If not, a new node re-selection is necessary.

The process of backtracking method to generate the path space is shown in Figure 3 [22]. First, one of the source and destination nodes (S, D) is selected as the initial expansion node. Regardless of any search process started at this initial expansion node, a reset on the path is is necessary, and the original expansion node is included in the path. Subsequently, from the set of neighboring nodes of the newly expanded node, a node is chosen as the next node based on a random selection process and then removed from the set to prevent redundant exploration. Finally, for the new node, it is applied to the judgment of the constraint function. For a given node, it is first determined whether it satisfies the constraints. If it is satisfied, it is necessary to further confirm whether the node is a goal point. If it is, it is included in the path list and the acquired path information is stored in the jump space. Conversely, if the path is not a goal point, then it will be considered as a new extended node and the probing of it will be started. However, if the new node fails to satisfy the constraints, we will return to the previous extended node and then search and probe for new nodes in the cluster of neighboring nodes around the new node.

doi:10.20944/preprints202503.0526.v1

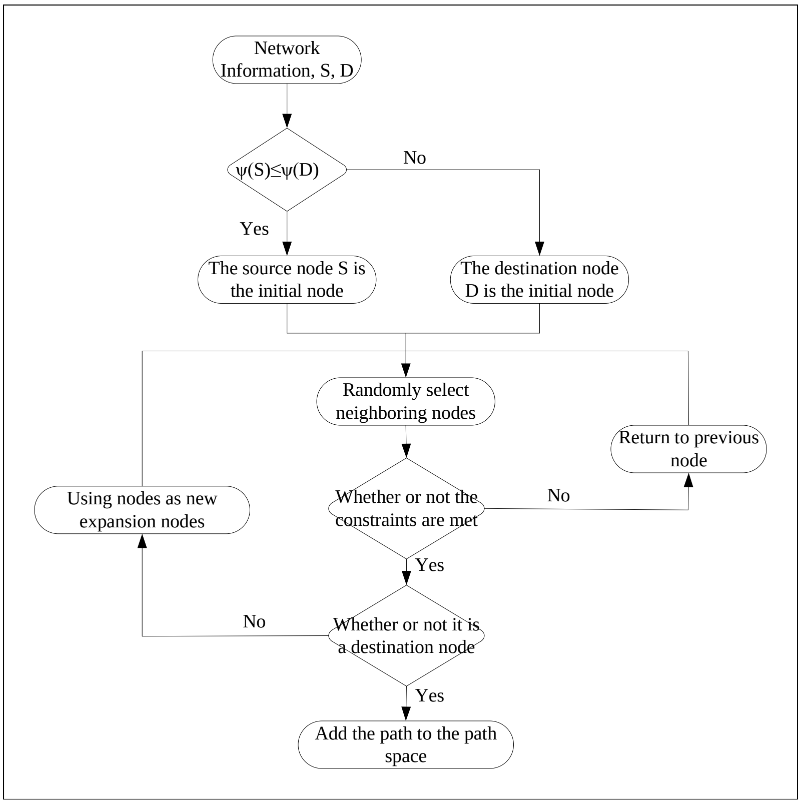
Network Information, S, D
ψ(S)≤ψ(D)
No
Yes
The source node S is the initial node
The destination node D is the initial node
Randomly select neighboring nodes
Return to previous node
Using nodes as new expansion nodes
Whether or not the constraints are met
No
Yes
No
Whether or not it is a destination node
Yes
Add the path to the path space

Figure 3. Generation of path space utilizing the backtracking approach

4. Experiments

4.1. Experimental Environment

In order to confirm the protection effect of the device address variable anonymity protection method based on address hopping, this paper employs the Mininet network simulation platform and the Ryu controller to build a simulated network topology. Additionally, the miniedit visualization tool bundled with Mininet is utilized to visualize the topology, as depicted in Figure 4, in which s1 to s11 are the Openflow1.3 protocol-supporting Openflow switches from s1 to s11 support Openflow1.3 protocol, c0 is the Ryu controller, and h1 and h2 are the communication hosts.

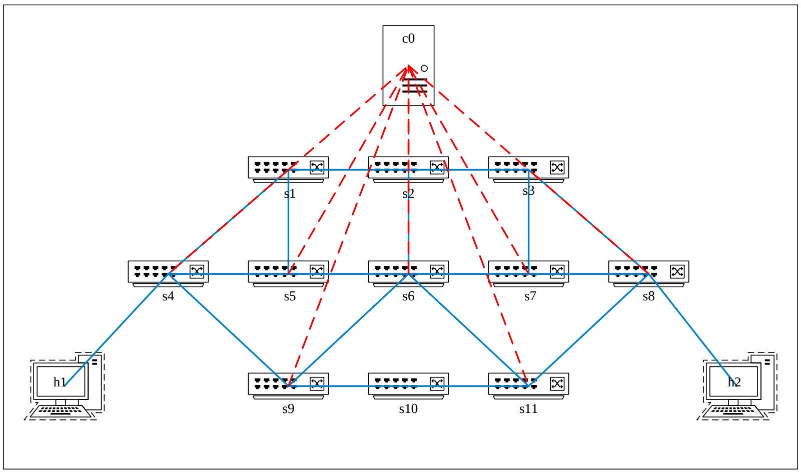

Figure 4. Experimental network topology diagram

4.2. Validity Analysis

In this paper, we use Namp to perform 100 random scans of the network, take the average value after integrating the scanning results, and analyze the results to find that the hosts in the network without device address hopping are all scanned before 70min, while only 43% of the hosts in the network using device address hopping are scanned on average. The probability of a host's real IP being hit by the scanner decreases significantly after the use of device address hopping, demonstrating the effectiveness of the algorithm against attacks, as shown in Figure 5.

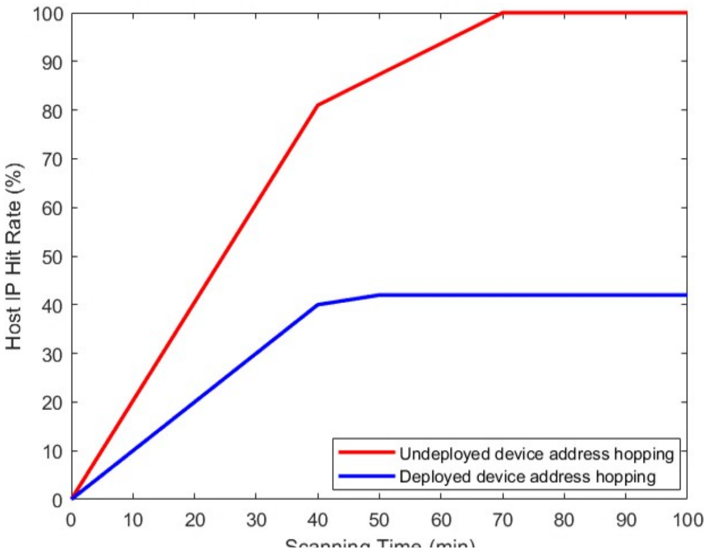

Figure 5. Comparison of host IP hit ratios with and without deploy device address hopping

The success rate of eavesdropping attacks correlates directly with the quantity of data stolen by the attacker. In this paper, we evaluate the effectiveness of the device address hopping algorithm by comparing the amount of data that can be obtained by the attacker in the two cases of whether or not device address hopping is deployed. In the simulation environment, the link bandwidth is set to 100Mb/s, Host1 sends packets to Host2 at a transmission rate of 10Mb/s, and the rest of the hosts in the network provide normal communication traffic. The hopping space between Host1 and Host2 is shown in Table 1.

Table 1. Device hopping information

Route Name	Route Details
Link1	s4-s1-s2-s3-s8
Link2	s4-s5-s6-s7-s8
Link3	s4-s9-s10-s11-s8

Using the Wireshark packet capture tool, we simulate an eavesdropping attacker to intercept packets transmitted through three different routes, and document the quantity of data transferred between Host1 and Host2 via each route every minute.

Without deploying device address hopping, data will be transmitted on the same fixed path. As shown in Figure 6, when no device address hopping is deployed, the communication path is random, and there will be a situation where a path is selected multiple times, such as in the 4th and 5th minutes, when the data forwarding of a single path is more than 50%, and if an eavesdropper places a wiretap on any node along this path, they would gain access to all session data exchanged between the communicating parties. If an attacker chooses to eavesdrop on this path, they will be able to access more than 50% of the communication data during this time.

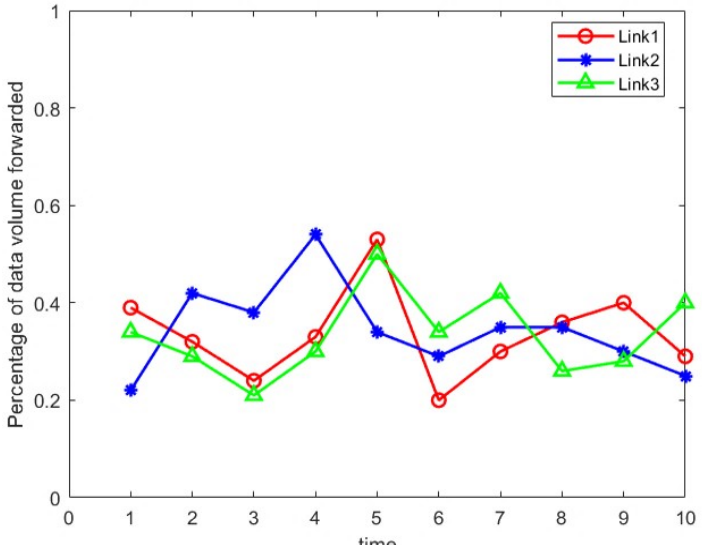

Figure 6. Data forwarding volume without deploy device address hopping

In the case of deploying device address hopping, as shown in Figure 7., as the communication between Host1 and Host2 continues, the amount of data transfer on different paths gradually equalizes, with each path accounting for about 30% to 40% of the total data transfer. In this case, if an eavesdropper is eavesdropping at any node in the network, they can only access at most 40% of the total data volume.

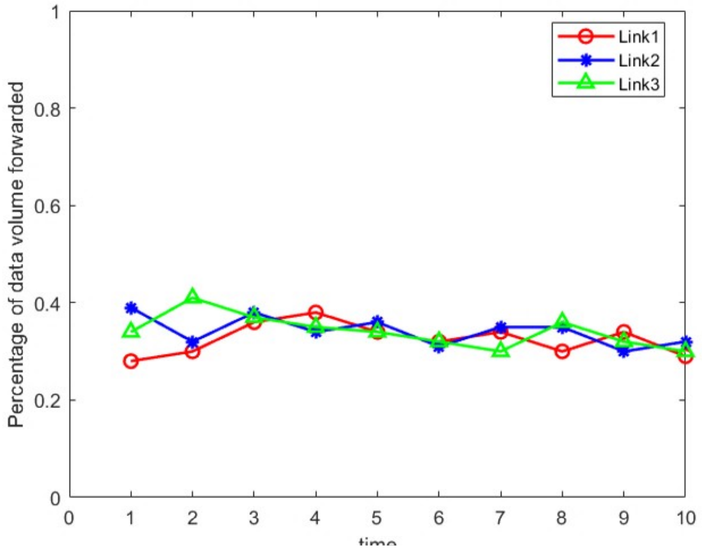

Figure 7. Data forwarding volume with deploy device address hopping

4.3. Availability Analysis

In this paper, we collect the delay data of Host1 and Host2 by performing connectivity tests on them, whether they use the device address hopping protection policy when communicating or not. This paper also compares the communication delay when the protection policy is enabled or not under different attack strengths. The specific experimental outcomes are presented in Figure 8.

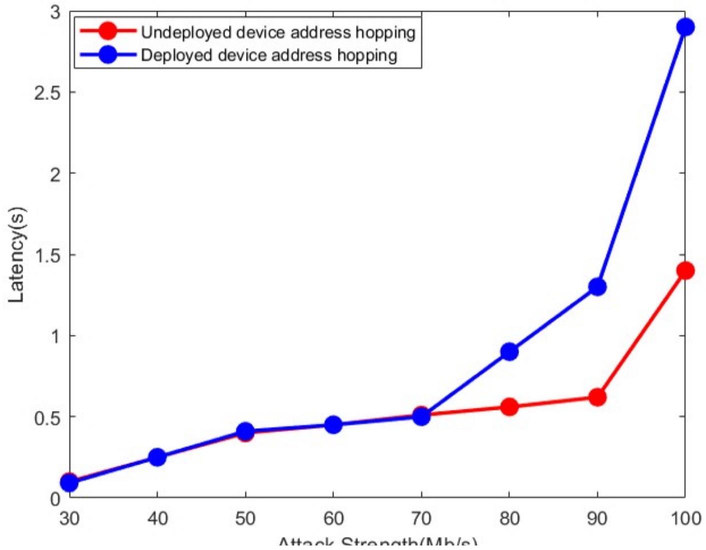

Figure 8. Comparison of the latency with or without deploy device address hopping

Without the device address hopping protection policy, the communication delay between Host1 and Host2 increases exponentially with increasing attack strength. When there is a device address hopping protection policy, since the communication path is randomly selected, there is a probability close to 1/3 to select the target path. Under this mechanism, the communication delay increases with the increase of the attack strength, but the communication delay is reduced by about 60% compared to the unprotected strategy.

Since the end information hopping requires downstreaming the flow table and updating the true/false end information mapping table, when the hopping frequency is reduced (the hopping period becomes longer), the CPU load that needs to be occupied is also lower. The comparison of CPU load under whether or not to adopt the device hopping protection strategy is shown in Figure 9.

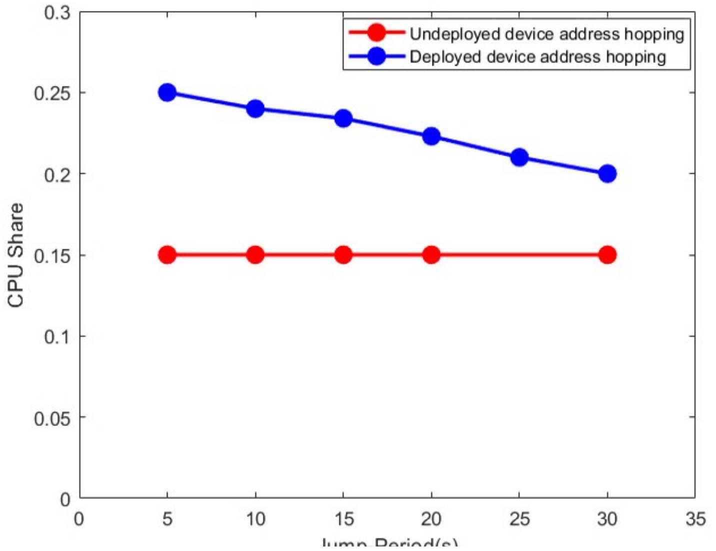

Figure 9. Comparison of the proportion of CPUs with and without deploy device address hopping

5. Conclusions and Future Work

The proposed address-hopping-based device identity anonymity protection method, which utilizes the address-hopping strategy in the mobile target defense (MTD) technology, significantly improves the defense capability and security of the network by dynamically adjusting the network topology and path strategy. First, the virtual network topology is randomly generated based on the collected node information and stored in the random network topology hopping pool. Then, CVSS is used to assess the vulnerability of the fake nodes and measure the structural stability and defense capability of the virtual network through quantitative analysis. Subsequently, the backtracking method is utilized to construct the hopping path space to ensure that the paths meet the criteria in terms of capacity, transmission delay and reachability, and at the same time, the non-overlapping constraint is set to enhance the diversity and security of the paths. Thereby, the anonymity of network devices is realized under the premise of satisfying specific constraints.

References

1. Li, Z., Zhang, Z., Fu, M., et al. A novel network flow feature scaling method based on cloud-edge collaboration. *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, **2023**, 1947–1953.
2. Li, Z., Wang, P., Wang, Z., et al. Flowgananomaly: Flow-based anomaly network intrusion detection with adversarial learning. *Chin. J. Electron.* **2024**, *33*(1), 58–71.
3. Sun, L., Xue, Y., Dong, Y., et al. An Novel Hybrid Method for Effectively Classifying Encrypted Traffic. *IEEE Global Telecommunications Conference*, **2010**, 1–5.
4. Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., et al. Characterization of encrypted and vpn traffic using time-related. *Proceedings of the 2nd international conference on informati on systems security and privacy (ICISSP)*, sn, **2016**, 407–414.
5. Yamansavascilar, B., Guvensan, M. A., Yavuz, A. G., et al. Application identification via network traffic classification. *2010International Conference on Computing, Networking and Communications*, **2017**, 843–848.
6. Wang, Z. The applications of deep learning on traffic identification. *BlackHat, USA*, vol. 24, **2015**.
7. Wang, W., Zhu, M., Zeng, X., et al. Malware traffic classification using convolutional neural network for representation learning. *International Conference on Information Networking*, **2017**, 712–717.
8. Zou, Z., Ge, J., Zheng, H., et al. Encrypted traffic classification with a convolutional long short-term memory neural network. *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems IEEE*, **2018**, 329–334.
9. Yao, H., Liu, C., Zhang, P., et al. Identification of Encrypted Traffic Through Attention Mechanism Based Long Short-term Memory. *IEEE Trans. Big Data* **2022**, *8*(01), 241–252.
10. Tong, X., Tan, X., Chen, L., et al. BFSN: A Novel Method of Encrypted Traffic Classification Based on Bidirectional Flow Sequence Network. *2020 3rd International Conference on Hot Information Centric Networking (HotICN)*, **2020**, 160–165.
11. Lu, B., Luktarhan, N., Ding, C., et al. ICLSTM: Encrypted Traffic Service Identification Based on Inception-LSTM Neural Network. *Symmetry* **2021**, *13*(6), 1080.
12. Manjunath, Y. S. K., Zhao, S., Zhang, X. P. Time-distributed feature learning in network traffic classification for internet of things. *2021 IEEE 7th World Forum on Internet of Thing s (WF-IoT)*, IEEE, **2021**, 674–679.
13. Cheng, J., Wu, Y., Yuepeng, E., et al. MATEC: A lightweight neural network for online encry pted traffic classification. *Comput. Netw.* **2021**, *199*, 108472.
14. Rui-Qin, H. Research on key technology of network layer mobile target defense based on SDN. *Strategic Support Force Information Engineering University*, **2022**. DOI: 10.27188/d.cnki.gzjxu.2022.000034.
15. Weizhen, H., Fucai, C., et al. Research progress of network layer-oriented dynamic hopping technology. *J. Netw. Inf. Secur.* **2021**, *7*(6), 44–55 doi: 10.11959/j.issn.2096-109x.2021104.
16. Yue-bin, L., Bao-sheng, W., et al. Akeyed-hashing based self-synchronization mechanism for port address hopping communication. *Front. Inform. Technol. Electron. Eng* **2017**, *18*(5), 719–728.
17. Haleplidis, E. Overview of RFC7426: SDN layers and architecture terminology. *IEEE Softwareization* **2017**.
18. Yuhang, W. Research and implementation of an SDN-based address hopping active defense technique. *Zhejiang University*, **2017**.
19. Chang, S. Y., Park, Y., Muralidharan, A. Fast address hopping at the switches: Securing access for packet forwarding in SDN. *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, IEEE, **2016**, 454–460.
20. Zheng, K., Zhao, X., Li, X., et al. A SDN-based IP address hopping method design. *2016 5th International Conference on Measurement, Instrumentation and Automation (ICMIA 2016)*, Atlantis Press, **2016**.

21. Hao, Z. Random routing moving target defense based on SDN. *North China University of Science and Technology*, **2023**.001127.
22. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*(15), 8121–8144.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.